(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 505 175 B1

(12) EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
10.02.1999 Bulletin 1999/06

(51) Int Cl.[6]: **G06F 7/52**

(21) Application number: **92302363.4**

(22) Date of filing: **19.03.1992**

(54) **Preprocessor of a division device employing a high radix division system**

Vorprozessor für Dividierer unter Verwendung eines Divisionssystems mit hoher Grundzahl

Préprocesseur pour diviseur utilisant un système de division à base élevée

(84) Designated Contracting States:
DE FR GB

(30) Priority: **19.03.1991 JP 78265/91**

(43) Date of publication of application:
23.09.1992 Bulletin 1992/39

(73) Proprietor: **FUJITSU LIMITED**
Kawasaki-shi, Kanagawa 211 (JP)

(72) Inventor: **Ito, Eiki, c/o Fujitsu Limited**
Kawasaki-shi, Kanagawa, 211 (JP)

(74) Representative: **Billington, Lawrence Emlyn et al**
Haseltine Lake & Co.
Imperial House
15-19 Kingsway
London WC2B 6UD (GB)

(56) References cited:
- **IEEE TRANSACTIONS ON COMPUTERS vol. 39, no. 12, December 1990, NEW YORK US pages 1424 - 1433 CARTER ET AL. 'Radix-16 signed-digit division'**
- **IBM TECHNICAL DISCLOSURE BULLETIN. vol. 23, no. 1, June 1980, NEW YORK US pages 249 - 251 MAASS 'Alignment of operands before and after 2-bit binary divide loop'**

EP 0 505 175 B1

Description

[0001] The present invention relates to a preprocessor of a division device, and in particular, to a preprocessor of a division device employing a high radix division system capable of producing a partial quotient "n" bits (n≧ 2) at a time.

[0002] A division operation is largely classified into three processes, namely; preprocess, main process, and postprocess. The preprocess is used to modify operand data into a form such that the main processor accepts the operand data, and the main process actually produces the quotient and a remainder. The postprocess modifies the output of the main processor to obtain a specified form suitable for a result.

[0003] The preprocess of the division device has various forms, depending on the systems of the main processes. The most widely known system of the main processes is a means such that one bit of partial quotient is obtained by subtracting a divisor from a partial remainder, the result of such subtraction of the original partial remainder is shifted by one bit, and the resultant value is made a new partial remainder (hereinafter referred to as one-bit system). The high radix division system is used to produce an n-bit partial quotient by a one time processing, in comparison with the one-bit system.

[0004] The one-bit system performs a normalization shift as a preprocess to produce a significant bit of "1" both for a divisor and a dividend. Such a processing is carried out by a preprocessor.

[0005] In the preprocessor of the division device as hereinbefore described, a drawback often arises when a high radix division requires both a quotient and a remainder at the same time. This is because an enumeration for obtaining a quotient must be stopped at the predetermined digit to leave a remainder, but such a predetermined digit is often not the last digit of group "n" digits.

[0006] Such a problem arises in the high radix division, and therefore, conventionally a dividend shift is performed after amending a micro program to make the bit number of the quotient to be obtained an integer multiple of "n".

[0007] However, such an amendment by the microprogram requires multiple machine cycles for the processing, and therefore, results in the deterioration of a division performance.

[0008] Such an algorithm is disclosed in IBM technical disclosure bulletin, Volume 23 No. 1 of June 1980, pages 249 to 251, in an article by K.K. Maass entitled "Alignment of operands before and after 2-bit binary divide loop". The divisor is first digit normalised, and then bit normalised. The dividend is then left shifted by the number of leading digits of the divisor, right shifted 4-bits, and then left shifted by the number of leading zero bits of the divisor. After division, a number of steps are taken to obtain the remainder from the result.

[0009] In "Radix-16 Signed-digit division", by T M Carter and J E Robertson, IEEE transactions on computers, Volume 39 No. 12 of December 1990, New York, US, another division algorithm is described, for use with digits having a maximum value of ± 10 and a radix of 16. Some numbers have redundant representations. Normalisation takes place to a particular value range, not to a particular digit in the leading position.

SUMMARY OF THE INVENTION

[0010] An object of the present invention is to provide a preprocessor of a division device capable of preprocessing a high radix division at a high speed.

[0011] According to the present invention there is provided a division device preprocessor as defined in claim 1.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] The present invention will be more clearly understood from the description as set forth below with reference to the accompanyinq drawinqs wherein:

Fig. 1 is a schematic block diagram explaining a conventional preprocessor of the division device;
Fig. 2 shows an operation of the conventional preprocessor of the division device;
Fig. 3 shows an operation of the conventional preprocessor of the division device;
Fig. 4 shows an operation of the conventional preprocessor of the division device;
Fig. 5 is a clock diagram showing an embodiment of a preprocessor of a division device according to the present invention;
Fig. 6 is a block diagram showing a division device using a preprocessor of Fig. 5 according to the present invention;
Fig. 7 shows an operation of a preprocessor of a division device according to the present invention; and
Fig. 8 shows an operation of a preprocessor of a division device according to the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0013] Before describing the preferred embodiments, an explanation will be given of the conventional preprocessor of a division device, with reference to Fig. 1.

[0014] Figure 1 is a schematic block diagram explaining a conventional preprocessor of the division device of a one bit system.

[0015] The one-bit system performs a normalization shift as a preprocess to produce a most significant bit of "1" both for a divisor and a dividend.

[0016] Such a processing is carried out by a preprocessor as shown in Figure 1. The preprocessor is comprised of a selector 11 for selecting a divisor or a dividend to be input, a zero counter 12 for counting the number of binary digits on which "0s" at the heads of a divisor and dividend are written, and a shifter 13 for shifting zero bits obtained by such counting to produce the most significant bit of "1". Producing the most significant bit of "1" is called normalization. The preprocessor outputs the number of "0" expression bits measured by a zero counter 12 and shifted by a shifter 13, and the normalized divisor and dividend, and sends the same to the main processor 10.

[0017] In such a process, "0111" + "0011" can be computed, for example, by the following procedure. A divisor Y=0011 is normalized to produce "1100" and a dividend X=0111 is normalized to produce "1110". At this time, assume that the shift amount (=sa) of Y and X are saY and saX, and saY=2 and saX=1 is produced, since the divisor Y=0011 is shifted two bits to "1100" and the dividend X=0111 is shifted one bit to "1110".

[0018] In the main processor, in Fig. 2, if a division is performed as a binary digit integer and a left bit of a dividend X is the 0th bit, then a quotient is produced only from the third bit (saY-saX+1), in this example producing only 2 bits. Further (3-saY+saX) pieces of "0s", in this example two "0s" are inserted to the left of the third bit, thus leaving a quotient " 0010".

[0019] A remainder starts at the (-saX)-th bit, and in this example "0001" corresponding to as much as 4 bits from the (-1)-th bit is left.

[0020] In the preprocessor of the division device as hereinbefore described, a drawback often arises when a high radix division requires both a quotient and a remainder at the same time, because an enumeration for obtaining a quotient must be rounded up at a predetermined digit to leave a remainder, but such a predetermined digit is often not the last digit of group "n" digits.

[0021] An example of such a drawback is shown in Fig. 3, where 1011 ÷ 0011 is computed, by using the high radix system capable of obtaining two quotient bits producing basis at a time. If producing quotient bits is started at the third bit, (saY-sax+1)= 2 - 0 + 1 = 3 is the number of bits to be produced and it is desired to attain only the fifth bit after obtaining the third and fourth bits at the same time by the high radix system. If simultaneously attaining the fifth and sixth bits, the quotient of the sixth bit is established. To obtain only a quotient, a rounding down of the sixth bit is preferred, but, to obtain a remainder, an amendment is required because of an excessing high minus of as much as the quotient of the sixth bit thereafter.

[0022] This example is satisfactorily processed when a quotient is attained from the second bit, but upon computing 1011÷ 0110, as shown in Fig. 4, the enumerating for a quotient is processed up to the fifth bit when the enumeration is intended to stop at the fourth bit.

[0023] Such a problem arises in the high radix division, and therefore, conventionally a dividend shift is performed after amending a micro program to attain "n" as the bit number of the quotient to be obtained

[0024] However, such an amendment by the microprogram requires multiple machine cycles for the processing, and therefore, results in the deterioration of a division characteristic.

[0025] An embodiment of a preprocessor of a division device in accordance with the present invention is described with reference to the drawings. Figures 5 to 8 show an embodiment of a preprocessor of a division device in accordance with the invention. In the present embodiment, a preprocessor of a division device is comprised of, as shown in Figure 5, a first selector 6 for selecting either a divisor or a dividend respectively to be input; a first zero counter 1 for counting the unit number of continued "0" bits when using "n" bits as a unit from among the continued "0" bits at the heads of the divisor and the dividend respectively input; a first shifter 2 for shifting the divisor and the dividend by a "n" bit unit based on the above counted value, a second zero counter 3 for counting the number of head expression "0" bits of the divisor having completed the first normalization (hereinafter referred to as "bsaY", standing for bit shift amount of Y); a latch 4 for storing the value of the zero counter, a second selector 7 for selectively outputting an output of the second zero counter 3 or an output of the latch 4, and a second shifter 5 for receiving the output of the second selector and executing a shift of the divisor by less than "n" bits and another shift of the dividend by the number stored by the latch 4. The preprocessor of the division device is used to output the normalized dividend X and divisor Y and the respective shift numbers.

[0026] Therefore, according to the embodiment, for example, when computing the high radix division with a radix 3 for 000 010 111 010÷ 000 000 001 001, the first selector 6 selects a divisor 000 000 001 001 and feeds it to the first zero counter 1. The first zero counter 1 counts the unit number of "0" bits existing from the head by using n=3 as a unit (hereinafter referred to as "nsaY", standing for n-bit shift amount of Y), and obtains the shift unit value nsaY=2 to be fed to the first shifter 2.

[0027] Similarly, the number of continued three "0s" bits "000" is counted when inputting 000 000 001 111, and leaves nsaY=2.

[0028] The first shifter 2 shifts the divisor by (nx nsaY) bits, i.e., by 3× 2=6 bits, and leaves 001 001 000 000 to send out to the second zero counter 3. The second zero counter counts expression "0" bit of the divisor shifted by the first shifter, obtains the shift number 2, and outputs the same to the latch 4 and the second shifter 5. The second shifter 5 performs a 2-bit shift, and outputs 100 100 000 000 and the shift unit number nsaY=2 of the first time and the shift number bsaY=2 of the second time.

[0029] To normalize the dividend, the first selector 6 selects dividend 000 010 111 010, which is input to the first zero counter 1. The first zero counter 1 counts the shift unit number using n=3 as a unit to obtain the shift unir number nsaX=1, and feeds it to the first shifter 2.

Then, the first shifter 2 shifts the dividend by 3 bits to obtain 010 111 010 000, and feeds it to the second zero counter 3. At this time, the second selector 7 selects the latch 4 to obtain the shift number 2, and feeds it to the latch 4 and the second shifter 5. The second shifter 5 performs a 2-bit shift to output the dividend 01 011 101 000 000 and the shift unit number nsaX=1 of the first time.

[0030] As hereinbefore described, according to the present embodiment, the difference between the shift numbers of the divisor and the dividend is 8-5=3, which is a multiple of 3. This relationship is always established, the bit number of a quotient operated is a multiple of 3, and the division is stopped at the predetermined digit to thus obtain an exact remainder.

[0031] In more detail, the divisor is shifted by (3 × nsaY + bsaY) bits, the dividend is shifted by (3 × nsax + bsaY) bits, and thus, the difference between the shift numbers is (3 × nsaY + bsaY) - (3 × nsaX + bsaY) = 3 (nsaY - nsaX), thereby resulting in a multiple of radix n=3 of the present example.

[0032] An example of the division device using the preprocessor according to the present embodiment will be described. The preprocessor is the same as described above, and the explanation thereof is hereinafter omitted.

[0033] Figure 6 shows a division device using the preprocessor according to the invention. In the drawing, reference numeral 21 depicts a latch for storing a first shift amount nsaX of the dividend counted by the first zero counter 1; 22 depicts another latch for storing the first shift amount nsaY of the divisor; 23 depicts still another latch for storing the second shift number bsaY of the dividend and the divisor; 24 depicts a Y register for storing the normalized divisor; 25 depicts an X register for storing the normalized dividend; 26 depicts a subtracter for computing nsaX-nsaY; 27 depicts a main processor for enumerating a quotient and a remainder; 28 depicts a quotient register for storing a quotient enumerated from the main processor 27; 29 depicts a right shifter for right-shifting a quotient stored in the quotient register based on the value of the subtracter; 30 depicts another right shifter for right-shifting a remainder stored in the X register based on value of nsaX and bsaY, and 31 depicts a third selector for storing a remainder from the main processor to the X register.

[0034] The change of the normalized divisor in the process of division is explained in detail in USP 4,722. 069.

[0035] When the latches 21, 22, 23 and the registers 24, 25 store nsaX, nsaY, bsaY, X, and Y, the number of repeating times nsaX-nsaY+1 for a quotient enumeration is determined by computing nsaY-nsaX by the subtracter 26. According to this process, the main processor computes to obtain a quotient and an intermediate remainder on processing basis. The quotient and remainder are respectively stored in the quotient register 28 and the X register 25, and after the processes are completed a predetermined number of times. the quotient is stored in the quotient register 28 and the remainder is stored in the X register 25.

[0036] Since the quotient and remainder correspond to the normalized dividend and divisor, then by the respective right shifters 29 and 30, the quotient and the remainder are obtained respectively by right-shifting the quotient by [n× (nsaX-nsaY)+p] bits and by right-shifting the remainder by (n× nsaX+nsaY+q) bits, where "p" and "q" are constants determined depending on a scaling of the divisor and the dividend and other conditions.

[0037] A practical computation is described assuming n=2. Producing quotient bits starts at the second bit. In Fig. 7, when computing 1011 + 0011, because the divisor's first shift number nsaY=1 and the dividend's first shift number nsaX=0 and the divisor's second shift number bsaY=0, then the number of the operational times nSaY-nsaX+1 is equal to 2, the quotient is 0011, and the remainder is 0010.

[0038] Similarly, in Figure 8, when computing 1011÷ 0110. because the divisor's first shift number nsaY=0 and the dividend's first shift number nsaX=0 and the dividend's second shift number bsaY, then the number of the operational times nsaY-nsaX+1 is equal to 1, the quotient is 0001, and the remainder is 0101.

[0039] According to the embodiment, the difference between the shift numbers of the divisor and the dividend is always an integer multiple of "n"; a digit number of a quotient is also an integer multiple of " n"; a division may be stopped at the predetermined digit. and a desired remainder obtained simultaneously with a quotient.

[0040] Note that the shift value may be different if a bit position of a shift register, to which a dividend is stored, is different, but the substantial shift amount is the same when the bit position of the shift register is fixed, for example, when a left bit of a dividend X is the 0th bit.

[0041] As hereinbefore described, in accordance with the present invention, a divisor and a dividend measures the bit number of head expression "0" bits by the first zero counter using "n" bits as a unit, and is shifted by the first shifter by a unit of "n" bits. Concurrently, for the divisor shifted by the first shifter, the number of remaining head expression "0" bits thereof is measured by the second zero counter, and its head bits are normalized by the second shifter to express "1", and the dividend is shifted by the second shifter by the number of the head expression "0" bits of the divisor of the second zero counter stored by the latch, namely, by the shift number of the divisor. Thus, the difference between the shift numbers of the divisor and the dividend is always equal to an integer multiple of "n", a digit number of a quotient is also equal to an integer multiple of "n"; the division can be terminated at a predetermined digit, and a required remainder can be obtained simultaneously with the quotient. Since the preprocessor of the division device is realized by hardware, the division process can be executed at high speed with an improvement of the characteristic of a computer.

Claims

1. A division device preprocessor for use with a high radix division system capable of producing a partial quotient "n" bits (n ≥ 2) at a time, said preprocessor comprising;

   a first zero counter (1) connected to the operand inputs for counting the unit number of "n" bit units of continued "0" bits at the heads of a divisor and a dividend that have been input, and which outputs the respective input divisor unit number and input dividend unit number;
   a first shifter (2) whose data input is connected to the operand inputs and whose shift input is connected to the output of the first zero counter, for shifting the input divisor and the input dividend by the unit number counted by the first zero counter multiplied by "n" bits obtained from the divisor and the dividend;

   said preprocessor being characterised by further comprising:

   a second counter (3) connected to the output of the first shifter for counting the number of the continued "0" bits at the head of the shifted divisor output by the first shifter (2);
   a second shifter (5) whose data input is connected to the output of the first shifter and whose shift input is connected to the output of the second counter, for shifting sequentially the divisor and the dividend shifted by the first shifter (2) by the number of bits output from the second counter (3), whereby the preprocessor outputs a normalised divisor and dividend in which the difference between the shift numbers of the divisor and the dividend is always an integer multiple of "n", and a digit number of a quotient is also made an integer multiple of "n", so that a division may be stopped at the predetermined digit, and a remainder can be obtained simultaneously with a quotient.

2. A preprocessor as set forth in claim 1, further comprising;

   a latch (4) for storing the value output by the second counter;
   a first selector (6) having the divisor and dividend as inputs for outputting one of the divisor and dividend to the first zero counter and to the first shifter; and
   a second selector (7) having the latch and the second zero counter as inputs, for outputting one of the value being output by the second zero counter and the value stored by the latch to the second shifter.

Patentansprüche

1. Divisionsvorrichtungsvorprozessor zur Verwendung mit einem Divisionssystem mit hoher Radix, das "n" Bits (n ≥ 2) eines Teilquotienten auf einmal erzeugen kann, welcher Vorprozessor umfaßt:

   einen ersten Nullenzähler (1), der mit den Operandeneingängen verbunden ist, zum Zählen der Einheitszahl von "n"-Bit-Einheiten von kontinuierlichen "0"-Bits am Anfang eines Divisors und eines Dividenden, die eingegeben worden sind, und der die jeweilige Einheitszahl des eingegebenen Divisors und Einheitszahl des eingegebenen Dividenden ausgibt;
   einen ersten Verschieber (2), dessen Dateneingang mit den Operandeneingängen verbunden ist und dessen Schiebeeingang mit dem Ausgang des ersten Nullenzählers verbunden ist, zum Verschieben des eingegebenen Divisors und des eingegebenen Dividenden um die Einheitszahl, die durch den ersten Nullenzähler gezählt wurde und mit "n" Bits, die von dem Divisor und dem Dividenden erhalten wurden, multipliziert wurde;

   welcher Vorprozessor dadurch gekennzeichnet ist, daß er ferner umfaßt:

   einen zweiten Zähler (3), der mit dem Ausgang des ersten Zählers verbunden ist, zum Zählen der Anzahl der kontinuierlichen "0"-Bits am Anfang des verschobenen Divisors, der durch den ersten Verschieber (2) ausgegeben wurde;
   einen zweiten Verschieber (5), dessen Dateneingang mit dem Ausgang des ersten Verschiebers verbunden ist und dessen Schiebeeingang mit dem Ausgang des zweiten Zählers verbunden ist, zum sequentiellen Verschieben des Divisors und des Dividenden, die durch den ersten Verschieber (2) verschoben wurden, um die Anzahl von Bits, die von dem zweiten Zähler (3) ausgegeben wurde, wodurch der Vorprozessor einen normierten Divisor und Dividenden ausgibt, wobei die Differenz zwischen den Schiebezahlen des Divisors und des Dividenden immer ein ganzzahliges Vielfaches von "n" ist und eine Stellenanzahl eines Quotienten immer ein ganzzahliges Vielfaches von "n" wird, so daß eine Division an der vorbestimmten Stelle gestoppt werden kann und ein Rest gleichzeitig mit einem Quotienten erhalten wer-

den kann.

2. Vorprozessor nach Anspruch 1, ferner mit:

einer Verriegelung (4) zum Speichern des Wertes, der durch den zweiten Zähler ausgegeben wird;
einem ersten Selektor (6), mit dem Divisor und dem Dividenden als Eingaben, zum Ausgeben von einem des Divisors und des Dividenden an den ersten Nullenzähler und an den ersten Verschieber; und
einem zweiten Selektor (7), mit der Verriegelung und dem zweiten Nullenzähler als Eingaben, zum Ausgeben von einem des Wertes, der durch den zweiten Nullenzähler ausgegeben wird, und des Wertes, der durch die Verriegelung gespeichert wird, an den zweiten Verschieber.

Revendications

1. Préprocesseur de dispositif de division destiné à être utilisé avec un système de division dans une base élevée capable de produire "n" bits ($n \geq 2$) d'un quotient partiel à un moment donné, ledit préprocesseur étant caractérisé en ce qu'il comprend :

un premier compteur de zéros (1) connecté aux entrées d'opérande pour compter le nombre d'unités des unités de "n" bits des bits à "0" continus en tête d'un diviseur et d'un dividende qui ont été entrés et qui sort le nombre d'unités du diviseur entré et le nombre d'unités du dividende entré respectifs ;
un premier dispositif de décalage (2) dont l'entrée de données est connectée aux entrées d'opérande et dont l'entrée de décalage est connectée à la sortie du premier compteur de zéros, pour décaler le diviseur entré et le dividende entré du nombre d'unités comptées par le premier compteur de zéros multipliées par "n" bits obtenus à partir du diviseur et du dividende ;

ledit préprocesseur étant caractérisé en ce qu'il comprend, de plus :

un second compteur (3) connecté à la sortie du premier dispositif de décalage pour compter le nombre de bits à "0" continus en tête du diviseur décalé sorti par le premier dispositif de décalage (2) ;
un second dispositif de décalage (5) dont l'entrée de données est connectée à la sortie du premier dispositif de décalage et dont l'entrée de décalage est connectée à la sortie du second compteur, pour décaler, de manière séquentielle, le diviseur et le dividende décalés par le premier dispositif de décalage (2) du nombre de bits sortis du second compteur (3), de telle manière que le préprocesseur sorte un diviseur et un dividende normalisés dans lesquels la différence entre les nombres de décalage du diviseur et du dividende sont toujours un entier multiple de "n" et un nombre de chiffres d'un quotient devient également un entier multiple de "n", de sorte qu'une division peut être arrêtée au chiffre prédéterminé et un reste peut être obtenu en même temps qu'un quotient.

2. Préprocesseur selon la revendication 1, comprenant, de plus :

un dispositif de verrouillage (4) pour mémoriser la valeur sortie par le second compteur ;
un premier sélecteur (6) ayant le diviseur et le dividende comme entrées pour sortir l'un du diviseur et du dividende vers le premier compteur de zéros et vers le premier dispositif de décalage ; et
un second sélecteur (7) ayant le dispositif de verrouillage et le second compteur de zéros comme entrées, pour sortir l'une de la valeur sortie par le second compteur de zéros et de la valeur mémorisée par le dispositif de verrouillage du second dispositif de décalage.

Fig. 1

PRIOR ART

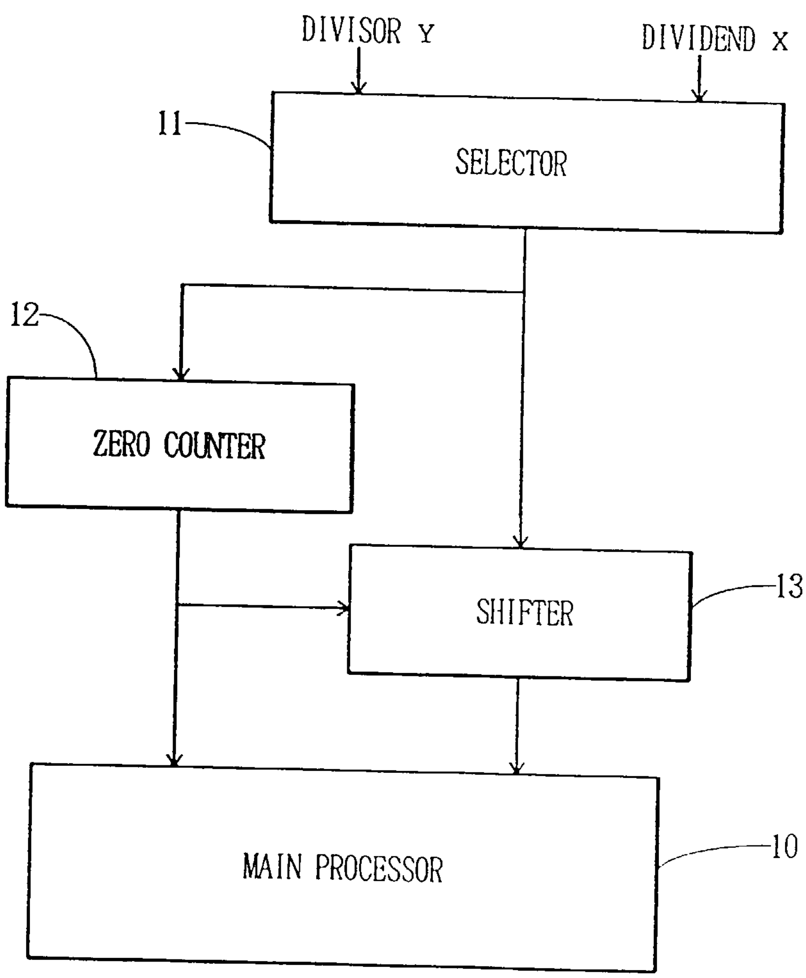


Fig. 2

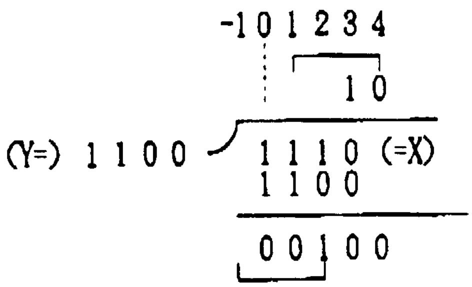


Fig. 3

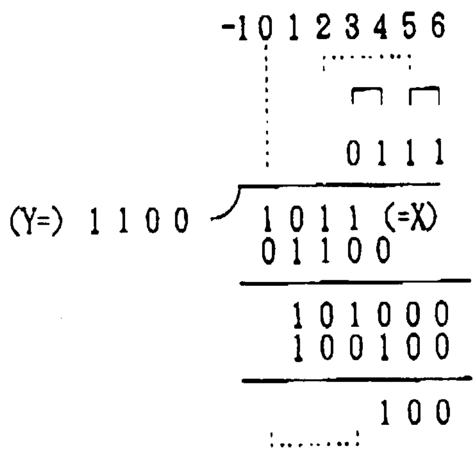


Fig. 4

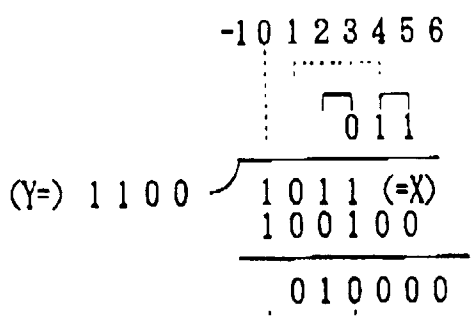


Fig. 5

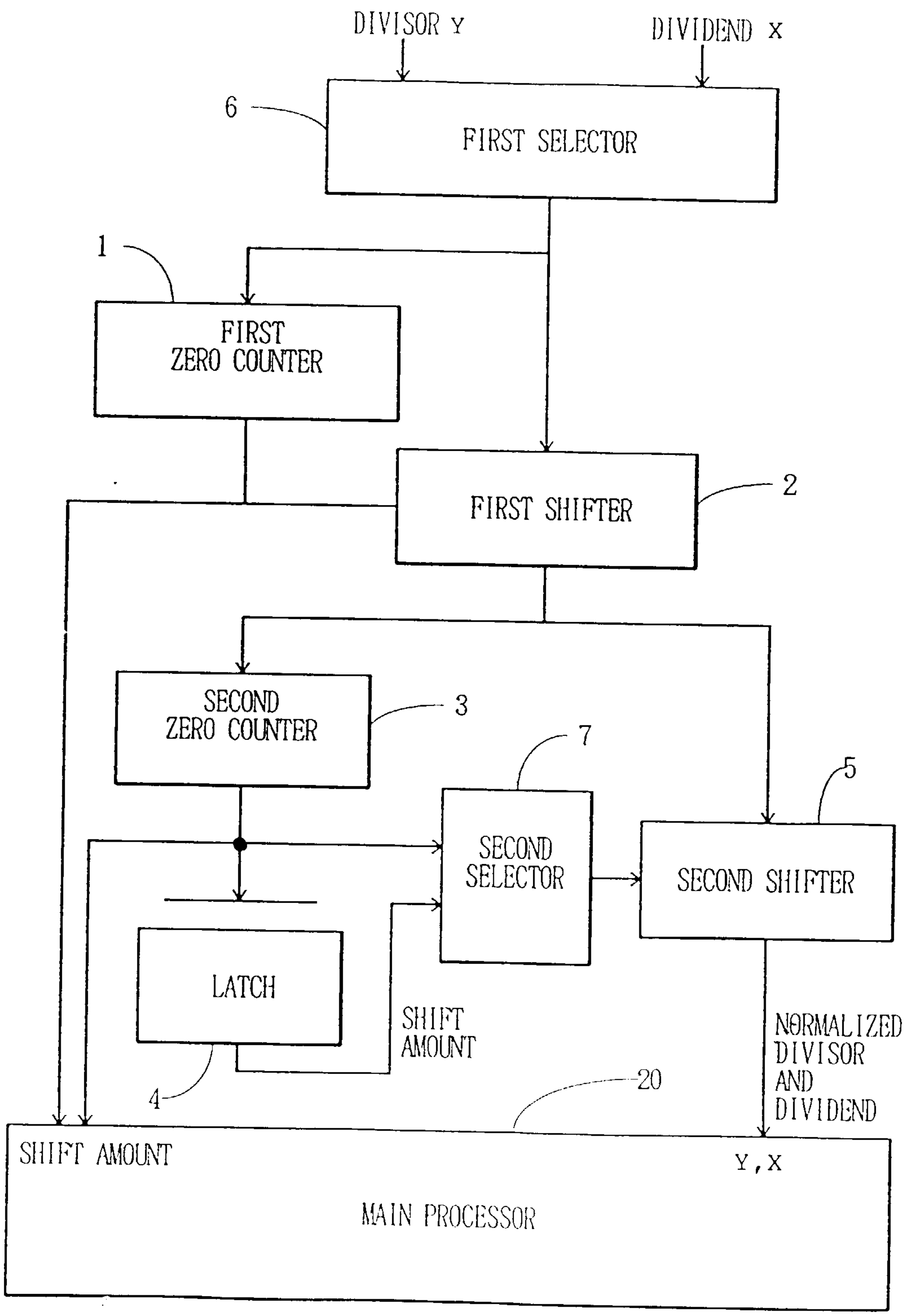

DIVISOR Y
DIVIDEND X
6
FIRST SELECTOR
1
FIRST ZERO COUNTER
FIRST SHIFTER
2
SECOND ZERO COUNTER
3
7
5
SECOND SELECTOR
SECOND SHIFTER
LATCH
SHIFT AMOUNT
NORMALIZED DIVISOR AND DIVIDEND
4
20
SHIFT AMOUNT
Y, X
MAIN PROCESSOR

Fig. 6

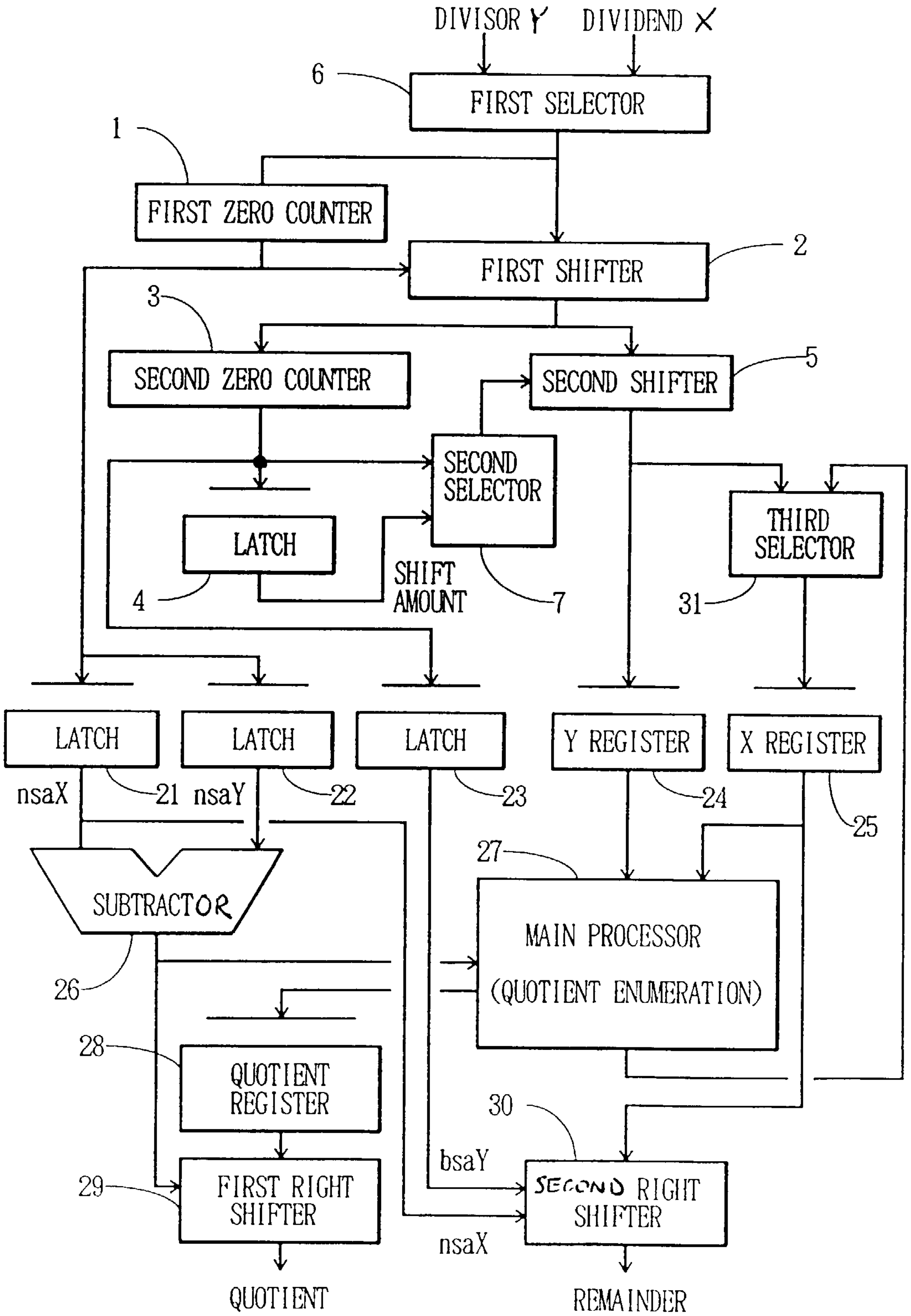

DIVISOR Y
DIVIDEND X
6
FIRST SELECTOR
1
FIRST ZERO COUNTER
FIRST SHIFTER
2
3
SECOND ZERO COUNTER
SECOND SHIFTER
5
SECOND SELECTOR
LATCH
SHIFT AMOUNT
4
7
THIRD SELECTOR
31
LATCH
LATCH
LATCH
Y REGISTER
X REGISTER
nsaX
21
nsaY
22
23
24
25
27
SUBTRACTOR
MAIN PROCESSOR
(QUOTIENT ENUMERATION)
26
28
QUOTIENT REGISTER
30
bsaY
29
FIRST RIGHT SHIFTER
SECOND RIGHT SHIFTER
nsaX
QUOTIENT
REMAINDER

Fig. 7

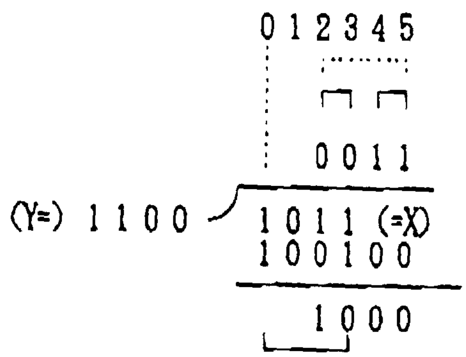
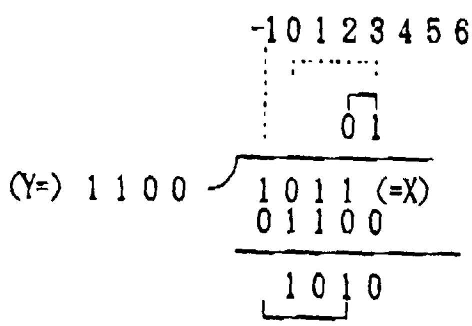

0 1 2 3 4 5
0 0 1 1
(Y=) 1 1 0 0
1 0 1 1 (=X)
1 0 0 1 0 0
1 0 0 0

Fig. 8


-1 0 1 2 3 4 5 6
0 1
(Y=) 1 1 0 0
1 0 1 1 (=X)
0 1 1 0 0
1 0 1 0